QUEUING METHODS FOR MITIGATION OF PACKET SPOOFING

RELATED APPLICATIONS

[0001] This application is a CIP of USSN 10/440,233 filed May 19, 2003.

FIELD OF THE INVENTION

[0002] This invention relates to communications networks and more particularly to methods and apparatus for mitigating service disrupting attacks such as denial of service (DOS) attacks in communications networks.

BACKGROUND

[0003] In communications systems such as those employing TCP/IP, data is transferred between end users via packets having a header which includes source and destination addresses. In a well behaved system the source and destination addresses allow a network user to communicate with and retrieve information from a server over the Internet. In the present description network users employ network devices which may be included in a local area network (LAN).

[0004] In recent years, malicious users of Internet services have been known to temporarily disrupt or even shut down Internet sites. This is typically done by taking advantage of inherent characteristics in the TCP protocol. For example, TCP uses a three-way handshaking protocol on connection set up. The handshake includes an acknowledgement message from the server to the user and one from the user to the server which confirms receipt of the message. An attacker is able to use a false source address (known as spoofing) which means that the server is unable to complete the acknowledgement portion of the protocol handshake. The server holds or stores incomplete or half opened connections for a period of time. During that time interval the attacker can flood the server and ultimately take the server out of service.

[0005] Similarly, an attacker wishing to disrupt an end user such as a user of a local area network can flood the LAN with multiple messages each having a phony or spoofed source address. Such an attack is known as a denial of service (DOS) attack which, ultimately, can shut down or deny service to the local area network.

[0006] Generally speaking a denial of service attack involves blocking a network user's ability to use some of the services provided by the network. DOS attacks are common across the Internet with many being launched daily at various targets. Many of the attacks involve specially constructed packets designed to either take advantage of flaws in the software or to tie up resources (resource flooding) within devices. The biggest obstacle in reacting to packet flooding attacks is the ability of the attacker to spoof i.e. disguise the source address of the packets.

[0007] Resource flooding attacks are effective when the attacker is capable of finding a bottleneck in the bandwidth or processing capabilities of a network device. The attacker floods the device with messages that congest the bottleneck and prevent legitimate requests from being processed.

[0008] For example, processing of messages may depend on data provided in the request (e.g., any identification or authentication data). The device may be required to perform a search in data structures (for example, a user database). For requests containing legitimate data, the time to search the data structure will tend to be fast (assuming a good database structure and implementation).

[0009] When a message contains information not found in the data structure (for example, an unknown user), the search time will be the longest possible for that structure. An attacker may take advantage of this by sending a flood

of requests that will require the maximum search time due to unsuccessful searches. Devices are not able to process legitimate requests due to the resources being consumed by the attacker.

[0010] Session Initiation Protocol (SIP) is a text based protocol similar to HTTP and SMTP for initiating interactive communication sessions between users. Such sessions include voice, video, chat, interactive games and virtual reality. SIP provides the necessary protocol mechanisms so that end systems and proxy servers can provide services such as IP telephony. An example SIP network is shown in Figure 6.

[0011] A SIP proxy server must search its user database for the user-ids found within SIP messages that it receives. An unsuccessful search for a user-id on the average takes much longer than a successful search. An attacker can cause a denial-of-service on the server by sending a flood of requests with invalid user-ids.

[0012] In the prior art, solutions have been proposed to mitigate the effect of computer viruses which search networks for vulnerable hosts. In a particular solution which is described, by Williamson M.M., in an article entitled "Throttling Viruses: Restricting propagation to defeat malicious mobile code", (June 17, 2002) packets with unknown destinations or hosts i.e. destinations or hosts that haven't been seen before, are subject to a series of timeouts that limits the rate of connections. This solution is host based using a mechanism designed to slow worm propagation. The above described solution examines the destination or host rather than the source addresses of packets and is not specifically designed to be network based.

[0013] Another prior art related to this invention has been presented by T. Peng, C. Leckie and K Ramamohanarao in an article entitled "Protection from Distributed Denial of Service Attack Using History-based Filtering" (presented May 14, 2003 but available earlier on the Internet). This solution is based on the notion of "good" and "unknown" source addresses. Under normal condition, their solution examines the source addresses of all IP packets. They keep the source addresses of all packets which appear more than k times (for some constant k). They also keep the source addresses of all packets which appear in at least d of the last n days (for some constants d and n). The source addresses fulfilling at least one of these two conditions define the "good" packets. Once a high-level network utilization that leads to packets being dropped is observed, this solution blocks any packets which do not have "good" source addresses. One major flaw of this approach is that it is effective only after a high bandwidth attack has been detected – therefore, an independent detection mechanism has to be provided. This may be useless for low bandwidth attacks like the TCP SYN flood attack. Another flaw of this approach is to partition the source addresses into only two categories.

SUMMARY OF THE INVENTION

[0014] The present invention relates to a mechanism for mitigating the affects of a packet flooding DOS attack by giving packet queue priority to clients which have been recognized as legitimate.

[0015] According to the present invention the packet queue priority technique is implemented in the network between a network device such as a LAN and the rest of the Internet and is designed particularly to mitigate DOS attacks on the LAN devices.

[0016] The present invention also provides a mechanism for mitigating the effects of data search resource exhaustion during a packet flooding DOS attack using request queuing priorities and data structure feedback.

[00017] In accordance with an aspect of the present invention there is provided an apparatus for providing priority queuing to packets at a network device in a communications network, comprising: a decision engine, at the network device, for receiving packets from the communications network and queuing each of the packets into an available queue wherein n queues shall be available and n • 2, in dependence upon a source address of the packet; and a scheduler for de-queuing packets from the queues for transmission to the network device wherein packets from the queues are de-queued at different rates depending on a level of trust associated to the source addresses. The higher the trust in the addresses the higher the rate at which the packets are de-queued from the given queue.

[0018] In accordance with a second aspect of the present invention there is a method of providing priority queuing to selected packets at a network device in a communications network, the method comprising: receiving packets from the communications network in a decision module at the network device; queuing each of the packets into an available queue wherein n queues shall be available, n • 2, in dependence upon a source address of the packet; and de-queuing packets from the queues for transmission to the network device wherein packets from the queues are de-queued at different rates depending on a level of trust associated with the source addresses. The higher the trust in the addresses, the higher the rate at which the packets are de-queued from the given queue.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] The invention will now be described in greater detail having reference to the attached drawings wherein;

[0020] Figure 1 is a high level illustration of the communication network of the present invention;

[0021] Figure 2 illustrates a physical embodiment of the solution;

[0022] Figure 3 illustrates traffic priority based on queuing decision;

[0023] Figure 4 illustrates connections with spoofed addresses;

[0024] Figure 5 illustrates established connections with constant address; and

[0025] Figure 6 shows an example of an SIP architecture; and

[0026] Figure 7 depicts the packet flows for a typical implementation of a second embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0027] Figure 1 illustrates, at a high level, elements of the present invention. Network devices such as a LAN shown generally by reference numeral 12 is connected to the network such as Internet 14. A traffic analyzer 16, which will be described in greater detail hereinafter, is implemented between the Internet and the network devices.

[0028] According to the invention the traffic analyzer 16, as shown in Figure 2, includes a decision engine 20, one or more source address tables 22, and a scheduler 24. According to the invention packets coming from the Internet 14 are monitored by the decision engine and the source addresses thereof are examined. If the source address is found in source address table labeled "Good" in Figure 2

the packet is designated high priority and scheduler 24 places the packet in a queue which is serviced at the highest rate. If the source address of the incoming packet is unknown i.e. not found in the good table it is placed in a queue which is serviced at a lower rate.

[0029] In Figure 3 it can be seen that incoming packets are classified into several classes of traffic based on analysis conducted by the decision engine. Multiple classes are defined in the present solution ranging from completely unknown clients to good clients. To move from one group to the next, a client must prove itself to be legitimate. In the present description only two classes (good and unknown/bad) are described but it will be apparent to one skilled in the art that the solution could be extended to multiple classes. It is also to be understood that according to an embodiment of the invention each of the multiple classes will have its own address table.

[0030] As indicated previously a table is created which stores the address information of clients that have been designated "good" for example. When a client transmits a packet, the good table is searched for the client's address. If found, the packet is placed in a higher bandwidth queue, and as a result, serviced at a higher rate. If the source address is not found within the good table then the packet is placed in a slow queue and the source address may be added to an "unknown/bad" table. It will be apparent that packets having source addresses that are stored in the "unknown/bad" table can be moved to the "good" table if it ultimately turns out that the packets are received from a legitimate source.

[0031] It is also within the scope of the present invention that clients having a source address that are known to be legitimate in advance can be pre-entered in the good table and, therefore, will always be given the highest traffic priority. Similarly, if clients having source addresses which have been established as

legitimate but ultimately proved not to be legitimate they will be removed from the "good" table.

[0032] To generate the good table each packet is examined. At the IP level it is possible to count the number of times that a source address has been observed. Once the source has been seen multiple times it is added to the good table. The exact number of times that a good source address must be seen before it is added to the good table is a implementation parameter of the system. This feature can be implemented with a counter for each address in the unknown/bad table.

[0033] Another selection criteria can be used for TCP packets. It is known that a TCP packet includes TCP/SYN packet at the beginning of a message and TCP/FIN at the end of a completed session. Since a TCP/FIN packet from inside the LAN indicates a successfully completed session, addresses from the FIN packets can be derived from the TCP/FIN messages and the addresses added to the good table. This could occur after one successful session or after several.

[0034] Figure 4 illustrates a packet flow for packets with source address previously unknown or which may contain spoofed IP source addresses such as would be found in a DoS attack. Since all of the incoming packets are not previously known containing source addresses of legitimate clients they are all placed sequentially in the slow queue. As indicated previously there may be multiple queues ranging from the fast queue to the slow queue.

[0035] Figure 5 shows the result of incoming packets in which the source address thereof has been moved from the unknown table to the good table during packet flow. As illustrated, in the initial stages packets marked D are placed in the source table for unknown addresses but as soon as a number of packets have been

examined and judged legitimate they are immediately sent to the queue having the highest priority.

[0036] The entries in the tables can be aged out so that only the most recent addresses remain or can be removed using a random early dropped (RED) algorithm. The length of time that entries would remain in the tables depends on traffic mode and the available table storage resources.

[0037] The RED algorithm is discussed in an article by Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413.

[0038] Using the solution of the present invention it makes it much more difficult for an attacker to successfully attack Network devices in the LAN using spoofed packets. Previously, the biggest difficulty in reacting to an attack stems from the fact that the attacker can insert any source address in their packets.

[0039] When the mechanism disclosed herein is implemented, the attacker must provide a legitimate, or stable, address (or successfully complete a connection) in order to have his address added to the "good" table. At this point the attacker can carry out a packet flooding attack, but all the packets must contain the same source address. This makes it possible to block packets from a specific address if it is determined that an attack is underway.

[0040] Legitimate users may see a slowdown for the first few packets of their connection, but then quickly will be upgraded to regular bandwidth and therefore should see little effect on their total bandwidth.

[0041] The solution presented herein may be less effective in situations where users make only one connection or short connections with long gaps in between. In those situations, no legitimate user ever stays on the "good" list long enough to gain the benefits of the high priority queue. In addition, a packet flooding attack will now fill the low priority queue and since the legitimate packets are considered as unknown as well they will be lost within the queue. It is possible to use a Random Early Drop algorithm on this queue to combat this disadvantage.

[0042] According to another embodiment of the present invention the mechanism takes advantage of the network devices ability to determine if a data structure search was successful. By modifying the queuing mechanisms described previously it is possible to mitigate both attacks causing random data structure searches and those with constant data searches. The concept of multiple queues and tables for unknown/bad traffic and good traffic previously described is used to moderate the messages being processed by the network device. Instead of using source IP address as the key for building the tables, the data that will be searched for within the network devices data structure is used. For example, a SIP proxy server 30, as shown in Figure 6, might build its table based on the user ID field of SIP messages.

[0043] The SIP architecture includes proxy servers 30 communicating with user agents 32 via the access networks 34. The proxy servers 30 communicate with each other via the Internet/Transport network. A location and registrar server 36 interacts with one or more proxy server 30.

[0044] With the methods previously described the denial of service due to a flood of attack messages in which the search data is randomized will be mitigated. Floods containing constant data may still be effective in causing a denial of service attack. This can be overcome by making a modification to the mechanisms used in

the tables thereby making use of some of the available application level information level. When an unsuccessful search occurs, the entries in the "unknown/bad" table and the "good" table, if necessary, are removed. This has the effect of ensuring that an attacker cannot use constant data to enter the fastest traffic queue. The solution, therefore, mitigates all denial of service attacks against network services due to data structure bottlenecks. The mechanism according to the present invention is shown graphically in Figure 7.

[0045] The solution proposed here presents a robust defense against wide classes of denial of service attacks exploiting higher that average search times when processing certain specific queries. It is applicable both to large devices with heavy, highly efficient data structures with fast access facilities and smaller devices with unsophisticated data organization. The solution mitigates attacks containing both messages with constant search information and those that are designed to cause random searches of the network device data structure. The previously described prior art solution would leave the network device vulnerable to constant data attacks.

[0046] Although specific embodiments of the invention have been described and illustrated it will be apparent to one skilled in the art that numerous changes can be made to the basic concept. It is to be understood, however, that such changes will fall within the full scope of the invention as defined by the appended claims.